THE OFFICE OF THE STATE CHIEF INFORMATION OFFICER
ENTERPRISE TECHNOLOGY STRATEGIES

North Carolina Statewide Technical Architecture

Conceptual Architecture

STATEWIDE TECHNICAL ARCHITECTURE

Conceptual Architecture:

Initial Release Date:	May 29, 2003	Version:	1.0.0
Revision Approved Date:	Not Applicable		
Date of Last Review:	March 11, 2004	Version:	1.0.1
Date Retired:			
Architecture Interdependencies:			
Reviewer Notes: Reviewed and updated office title and copyright date. Added a hyperlink for the ETS email – March 11, 2004.			

© 2004 State of North Carolina
Office of the State Chief Information Officer
Enterprise Technology Strategies
PO Box 17209
Raleigh, North Carolina 27699-7209
Telephone (919) 981-5510
ets@ncmail.net

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any informational storage system without written permission from the copyright owner.

Mission Statement

Conceptual Architecture provides a single, common, and cohesive vision that directs the design, construction, purchase, deployment, and management of information systems (I/S) and information technology (I/T) across state government.

Introduction and Background

Information technology can enable government to restructure operations and re-think service delivery mechanisms. These changes are necessary for government to work better, cost less, and become more citizen focused. However, the state's present technical infrastructure constrains its ability to improve the way it does business.

The state is confronting new and serious challenges: Citizen expectations for greater efficiency in operations and improved effectiveness of programs; implementation of new political initiatives; and competition from private enterprise for performing some services. State government must develop the capability to respond more quickly to a rapidly changing environment. The state's approach to the management and use of I/T is critical to successfully meet its expanding responsibilities, and to satisfy the increasing needs of its citizens.
The state's present technical architecture largely reflects the capabilities and applications systems design approaches of the 1970s and early 1980s. Although they served the state well in the past, these old systems are difficult to modify as business requirements change. It is also extremely expensive to expand them to provide additional required features and functions.

A new technical architecture is required to facilitate the improvement of business processes.

This document defines the new technical architecture. It serves as a blueprint for the design and deployment of the full spectrum of information systems owned by the State, and for the computing, communications, and management infrastructure required to support these systems. The Statewide Technical Architecture will allow individual departments to respond to specific business needs using common components, thus ensuring that information systems will be shared and managed on a statewide basis.

The new architecture must accomplish four major objectives.

- **It must support the business and program priorities of state government.** Technology investments must be channeled in areas that will give measurable improvements in public service. In particular, the new architecture must enable the development of systems that facilitate the implementation of new business processes and the creation of innovative service delivery approaches.
- **It must enable new applications to be developed faster and modified quickly, as business needs and program requirements change.**
- **It must simplify the support of operations, so that the state's technical infrastructure can be managed efficiently and reliably.** The new architecture will prescribe appropriate standards for technology. As a result, old and new systems will work together, and the greater use of common components, which will be shared on a statewide scale, will enable the infrastructure to be managed in a cost-effective manner.
- **It must enable agencies to continue to capitalize on the state's existing investment in applications and technology, as appropriate, while enabling a different approach to implementing systems.** New applications and enhancements to old systems will be built by the assembly of standard, modular and reusable components.

New systems must be developed to accommodate more rapid rates of change in the business and the technical environment. This approach is called "adaptive systems." **Adaptive systems are designed to be easily modified to facilitate changes in the business programs they support.**

The Statewide Technical Architecture defines the guidelines and standards enabling the state to implement and take advantage of adaptive systems.

A. Technology Components

The North Carolina Statewide Technical Architecture is comprised of a series of interrelated components. The twelve components of the architecture, in concert, provide the basis for the state to take advantage of adaptive systems in support of its business. (See Table 1-1).

Twelve Technical Architecture Components	
Application	Focal point for the state's systems inventory. It defines the way applications are designed and deployed, and how they cooperate.
Data	Provides high-quality, consistent data where it is needed to support business and transactional systems. It requires that data be accurate and easily accessible.
Groupware/Electronic Information	Provides the infrastructure for state staff to work collaboratively and share information electronically, regardless of where they are geographically located.
Information	Provides standards and guidelines for accessing data for decision support and analytical processing.
Integration	Enables applications in the state's inventory to cooperate and interoperate.
Middleware	Provides robust, standardized mechanisms by which applications can communicate over the statewide network.
Network	Defines processors interconnectivity. Provides the communication infrastructure for distributed applications and business locations.
Platform	Defines vendor-independent desktop and server components (hardware and operating systems) that support the architecture.
Componentware	Provides standards and guidelines that support reuse of robust application code for programmer productivity gains.
System Management	Identifies a systems management architecture that efficiently and effectively manages the state's distributed computing environment to support and enhance the productivity of its automated business systems.
Security and Directory Services	Identifies criteria and techniques associated with protecting and providing access to the state's information resources. It facilitates identification, authentication, authorization, administration, audit, and naming services.
Accessibility	Provides standards for accessing information by persons who must receive it in a form that is different from the manner in which it is normally presented.

Table 1-1. Twelve Technical Architecture Components

Subsequent sections of this document will describe each of the architectural components in detail. Where appropriate, the sections will provide standards to be followed by agencies in construction or procurement of technical components. Also, where appropriate, the sections will describe any statewide contracts in place to support the procurement of hardware or software.

B. Background

As a preliminary step in defining a Statewide Technical Architecture, the state subscribed to META Group, Inc.'s Enterprise Architecture Strategies (EAS). EAS is an advisory service of the nationally recognized META Group. It provides consulting and planning services, to both the public and private sectors, for re-engineering I/S architectures in order to implement client/server technology in a distributed systems environment.

The EAS framework provides a structure for North Carolina's specific architectural components and associated standards. This Statewide Technical Architecture document adds detail to this framework. The Conceptual Architecture chapter describes the foundation assumptions on which the 12 technical architecture components are built. The Conceptual Architecture drives the design and implementation of the 12 component architectures and provides logical consistency across those component architectures. (See Figure 1-1).

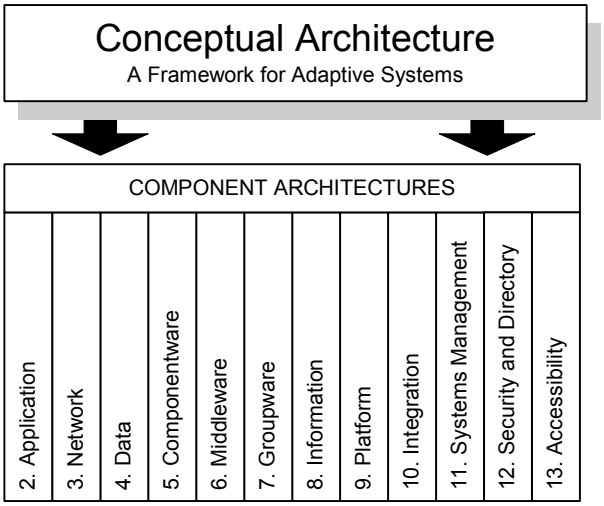

Figure 1-1. Conceptual Architecture

C. Strategic Objectives

The Statewide Technical Architecture Strategy adopted by the IRMC in December 1996 outlines a set of objectives for the technical architecture. Table 1-2 illustrates how the components of the technical architecture support those objectives. Note that some components of the architecture support more than one objective (e.g., System Management supports all the other components of the architecture) and that some objectives require support from more than one architectural component to succeed.

Strategic Objective	Architectural Component(s)
Modular designs	Application, Data, Componentware, Middleware, Accessibility
Component reusability	Componentware, Middleware, Application
Common presentations	Componentware, Middleware, Accessibility
Explore the potential for new technologies to	Groupware/Electronic Information,

Strategic Objective	Architectural Component(s)
enable better and faster service delivery	Platform, Network, Security and Directory
Share information among service providers	Information, Network, Integration, Groupware/Electronic Information
Technology standards	All components
Shared services	Groupware/Electronic Information, Componentware, Integration, Network, Security and Directory
Interoperable components	Componentware, Middleware, Application
Standards for components	Application, Componentware, Middleware, Systems Management, Platform, Network
Policies and procedures for operations	Systems Management, Security and Directory Groupware/Electronic Information
Reliability of components and operations	Componentware, Systems Management, Security and Directory
Standard interfaces	Application, Middleware, Componentware, Integration, Systems Management, Accessibility
Integrated, mutually supporting plans and implementations.	Application, Integration, Systems Management, Security and Directory
Maximize past resource investments while transitioning to the new architecture	Integration, Application, Information

Table 1-2. Strategic Objectives Supported by Architectural Components

III. Principles

The following principles are provided to guide the design and selection of technology components that will support the development of adaptive systems across the state.

Principle 1: Development cycle times are shrinking

The rate of change in the business and administrative process of organizations is accelerating. Consequently cycle times for implementing new service delivery mechanisms are shrinking.

Rationale:

- In the 1970's and 1980's cycle times were typically 7 to 10 years in length.
- In the 1990's, cycle times are averaging 12 to 24 months in length.

- Cycle time can sometimes be identified by examining the 'pay back' period required for most I/S projects -- particularly client/server (see Figure 1-2).

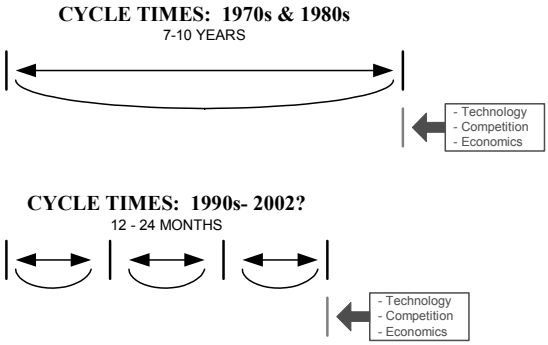

Figure 1-2. Cycle Times

Principle 2: Many business factors require shorter development cycle times

The accelerating rate of change and shrinking cycle time are exacerbated by a wide range of factors, including pressure for tax reduction, increasing demand for government services, and higher citizen expectations for customer service.

Rationale:

- Pressure for tax reduction limits funding availability for I/T investments.
- Legislative initiatives increase pressure for shorter payback periods for I/T investments.
- Citizen expectations for hours and types of service delivery are set by the private sector (e.g., bank ATM machines, 24-hour access to stores, on-line ordering of products, etc.)
- The amount of time spent by a citizen to access and receive services is factors in improving citizen satisfaction with government.

Principle 3: New applications must be implemented faster

The accelerating rate of change and shrinking cycle times require new applications to be implemented (and existing applications to be modified) at increasingly faster rates.

Rationale:

- Legislative changes and governmental initiatives involve shorter lead times for implementation. Examples of this include the new structured sentencing law and welfare reform.
- Cross-agency programs require the coupling of applications. Work First and new programs in child support (e.g., denying deadbeat parents driving privileges) are examples of this trend.

Principle 4: Existing infrastructure inhibits rapid change

The existing I/T infrastructure of organizations is inhibiting the ability to respond to shrinking cycle times.

Rationale:

- Most older technology and present applications had, as design assumptions, relatively static, slow changing, well planned, organizational environments.
- The perception of slow response to rapidly changing technology is the primary driver for individual program groups to bypass I/T organizations and implement their own local solutions.

Principle 5: Primary design point is to facilitate change

A single, consistent, primary design point for the statewide technical infrastructure must be implemented to facilitate change across the enterprise. The primary design point is to facilitate change in the business and administrative processes and the applications that enable them.

Rationale:

- Systems support business and administrative processes. If the business and administrative processes change then the systems that support those processes must be primarily designed for flexibility and adaptability.
- This requires that I/S staff design, implement, and manage for rapid change - applications, databases, server configurations, network designs, development methodologies, administrative processes, the I/S organization, support services, and so forth.

Principle 6: Business and IS staff must have a common vision

Adaptive systems require business and program organizations and I/S staff to share a common and cohesive vision of both the business and the role of technology in supporting the business. This requires I/S staff to have a significant knowledge of the business.

Rationale:

- Without I/S understanding the business, the business may have to change to adapt to the constraints of technology.
- Business units must also have a significant understanding of the technical architecture in order to redesign business processes to take advantage of the support and benefits technology can provide.

Principle 7: Business processes drive technical architectures

The technical architectures of adaptive systems are driven by the business processes of the enterprise (See Figure 1-3).

Rationale:

- Business and administrative processes drive the application architecture.
- The requirements of the application architecture drive the technical infrastructure.
- Adaptive systems are process-driven architectures.
- The encapsulation and management of process knowledge across the enterprise is critical.

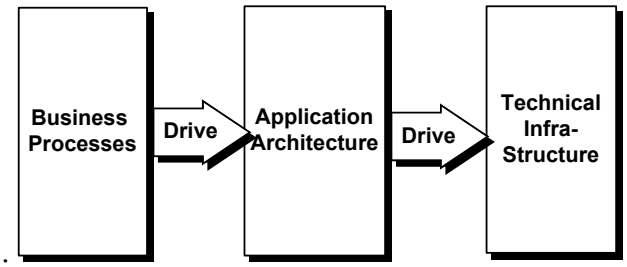

Figure 1-3. Technical Architectures are Driven by Business Processes

Principle 8: Technical infrastructure enables business changes

Rapid changes in business processes are enabled, in part, by implementing a technical infrastructure broader than the immediate application requirements.

Rationale:

- The rate of change may no longer give I/S staff the time to fully analyze, design, and implement the 'best' technical solution for a new application requirement.
- The value of future flexibility justifies the investment associated with creating the 'adaptive range' of process-driven architectures, rather than optimize on individual, point solutions. For example, if a proposed application must handle 1000 transactions per day this year and 5000 transactions per day in three years, it is better to invest in the capacity to support at least 5000 transactions per day, than to outgrow the system in three years and have to replace it.

Principle 9: Adaptive systems implementation must be statewide

Adaptive systems must be implemented as a core business strategy on a statewide basis, rather than on an individual agency or individual program basis. This requires a 'systemic vision' by all.

Rationale:

- There is a natural tension between the objectives of an individual agency or program and statewide objectives. Consequently, the best practices for implementing a specific application may be in conflict with the best practices for implementing statewide strategies. Adaptive systems demand an enterprise-wide vision for its success.

Principle 10: Information must be "pushed" to users

Organizations must evolve from a "pull" model of information access to a "push" model of information leverage.

In the "pull" model, the responsibility is on the service worker to determine the information required and request that information. In the "push" model, the system automatically notifies the appropriate user of a recent event (typically via electronic mail, paging, faxing, etc.) or provides the user with additional information that may be useful (e.g., providing a law enforcement officer with an arrest history during a traffic stop).

Rationale:

- As agencies think of more efficient ways to conduct business, we can take advantage of the benefits of new technology. This necessitates that both I/S staff, program managers, and end users think differently about how applications behave.
- Technology advancements have enabled this paradigm. In the past, the technology to support the push model was not available.

Principle 11: Technical architecture must be extensible and scaleable

The technical architecture must be extensible and scaleable across the enterprise in order to achieve adaptive systems.

- "Extensible" means the ability to easily integrate new technology and functionality.
- "Scaleable" means the ability to quickly meet demands for increased performance -- processing power, network connectivity, or data storage.

Rationale:

- Technology continues to advance at a rapid pace and the technical architecture must be able to accommodate these changes.
- When information becomes available in ways that make workers more productive, demand for that system will increase. It is important that the system is easily scaleable to meet those demands.

Principle 12: Adaptive systems concepts apply to purchased applications

Organizations should use adaptive systems concepts in the design of their technical architecture even if most applications are to be purchased.

Rationale:

- Without a technical architecture strategy, purchased applications significantly determine the architecture of the organization. Additionally, the result can be a technology and application base that is very difficult to integrate, chaotic to support, and unable to accommodate change in the business.

- It is easier to integrate a purchased application into an adaptive systems architecture than to try to modify an architecture designed around purchased applications.

Principle 13: Strong information delivery systems are required

There is extraordinary leverage in the data already owned by most organizations. Consequently, strategies that implement strong information delivery are required.

Rationale:

- Implicit in shrinking cycle times is the requirement for accelerated decision making.
- Like most organizations, the state has tremendous amounts of data, but we derive little useful information from it for conducting business, managing operations, and assessing program effectiveness.

IV. Recommended Best Practices

Recommended practices for developing adaptive systems are provided below.

Recommended Best Practice 1: Use partitioned application designs

Logically design application systems and databases to be highly partitioned with clear logical boundaries that are inviolate.

- Determining logical boundaries is extremely difficult and must be done in conjunction with users.
- Physical implementation (i.e. the platform architecture) is a secondary issue.

Recommended Best Practice 2: Design interfaces to be message-based

Design interfaces between application systems and subsystems to be message-based and adhere to the logical boundaries.

- Messages should request or spawn transactions.
- Messaging best preserves logical boundaries.
- Whenever possible, messages should be asynchronous in their logic.
- Whenever possible, messages should be asynchronous in their communication.

- Interfaces between application systems should not be at a file or database level; they should be implemented using messages.
- Preserving the logical boundary is extremely difficult and must be constantly tested in design reviews.

Recommended Best Practice 3: Design systems to be event-driven

Design application systems to be event-driven.

- Business events must immediately spawn an action(s) such as sending a message and/or initiating a transaction in real-time.
- 'Triggers' initiated by a database management system are a common approach. Agents and daemons are also good approaches.

Recommended Best Practice 4: Send messages in real-time

Sent transaction messages in real-time.

- Batch processes are not appropriate for sending transaction information from an 'upstream' application system (or subsystem) to the 'downstream' application system (or subsystem).
- The downstream application controls the rate or cycle of transaction processing (i.e., the downstream application may initiate a batch process to read from the message queue).
- Messaging subsystems provide flexibility for the application designer.

Recommended Best Practice 5: Use granular platform designs

Design platform design to be biased toward granularity in physical servers.

- Granularity in servers facilitates the partitioning of application systems and databases and the preservation of logical boundaries.
- Multiple, highly scaleable, mid-size servers that have binary compatibility provide more flexibility than a limited number of very large servers.
- Servers should be centrally managed as a multiple instance of a single image -- and servers should be located in data centers (which may be 'virtual' such as a secured closet).
- Granularity in servers significantly facilitates faster changes in business processes.
- Granularity in servers results in high up-front costs, and increased operational and support demands, but results in the lowest life-cycle costs in a period of change. The tradeoff is increased flexibility in the business for operational requirements.

Recommended Best Practice 6: Separate OLTP from EIS/DSS and OLAP

Physically separate transaction processing (OLTP) from EIS/DSS functions and on-line analytical processing (OLAP). This is the basis for an information warehouse strategy.

- Growth in OLTP is incremental in most organizations (8-12% per year). While the growth in user computing and user requests for data access are escalating (50-100% per year). If user processing is not separated it will begin to impact transaction throughput.
- Separation requires well-designed production data extraction, propagation, and replication to the information warehouse, often on a near real-time basis.
- Production data from applications should be loaded into the information warehouse for user reporting and analysis using standard tools (rather than specific tools of the application provider).

Recommended Best Practice 7: Implement client/server systems

Implement application systems using a client/server model in which a desktop processor (client) employs a graphical user interface (GUI) to share application processing with a server(s) over a LAN.

- This is a 'thin' client model. It improves system management since the application logic is on the server, not on every client.
- Design application systems for N-tier processing, but deploy the application on two physical platforms whenever possible. The application processing and database accesses should be on one physical platform and the user interface on the other physical platform.

Recommended Best Practice 8: Implement an enterprise-wide communication network

Implement an enterprise-wide, backbone communication network providing a single, network image to authorized users.

- The goal is to create an enterprise-wide, virtual LAN.
- An 'authorized user' may be an application system, a server, or a service layer.
- Requisite technologies include: TCP/IP, a high-speed network backbone, and powerful routing capability.

Recommended Best Practice 9: Redefine the role of the programmer

Redefine the domain of the programmer. This is necessary so that rapid, architected application development for both new requirements, as well as for changes to existing systems, can be achieved.

- This allows a division of labor, so that programmers will no longer have to be expert in all areas. Special programmers' skills can be targeted to specific types of programming:
  - User interface: implemented using prototyping and GUI tools.
  - Business rules: implemented using C++, Java, COBOL, or a 4GL.
  - Data access: implemented using SQL.

Recommended Best Practice 10: Use granular application designs

Design business rules and other recurring application logic in a consistent manner, encapsulated in a highly granular form, network based, and available across the enterprise under the control of a designated owner.

- These business rules can be implemented in a variety of ways: stored procedures using proprietary 4GL of database vendors, C++ and COBOL subroutines, objects and components.
- Strive for granularity in server resident business rules with a goal of 1 function point each.
- The expression 'server resident' describes the physical location of the business rules; however, from a logical perspective, the business rules simply reside on the network.
- Server resident business rules must be centrally managed and maintained.
- Server resident business rules require an enterprise-wide strategy.
- Server resident business rules are developed by separate programmers than those who develop the presentation and GUI components.
- Server resident business rules are the focal point of reusability -- no programmer should ever code a business rule that already exists anywhere in the enterprise.

Recommended Best Practice 11: Provide an information warehouse

Provide an information warehouse in order to facilitate end user data access and reporting, and to improve decision-making.

- The information warehouse substantially contributes to the change in the domain of the programmer.

Recommended Best Practice 12: Measure the contribution of I/S

Implement metrics, via the I/S organization, measuring the contribution of I/S to achieving the objectives of the business.

Recommended Best Practice 13: Centralize architecture management

Centralize the architecture management even if application systems are implemented locally.

- Much of the rigor and discipline of traditional systems should be maintained.

Recommended Best Practice 14: Implement open systems

Implement a consistent architecture, based on product, market, and industry standards, in order to achieve the objectives of open systems.

- Open standards do not exist for all parts of the architecture. Therefore, a combination of de facto industry standards, product standards, and open standards will be required in order to support a heterogeneous operating environment.
- Standards must be enforced by line managers who understand the importance of consistency in order to facilitate change. Standards cannot be successfully enforced by I/S alone.

V. Implementation Approach

Table 1-3 describes the current Statewide Technical Architecture implementation initiatives. Following the table, detailed information is provided projects that were approved as of November 1997.

Architecture	Project Name	Project Description	Status as of 07/ 2000
Conceptual	Application Portfolio Management System	Provide a system for managing applications as a portfolio of assets, and provide support for the state's technology planning process.	Approved and in progress with 12 of 24 agencies implemented
Data	Federated Metadata Repository	Establish and populate a repository that documents statewide data element standards. Enable state agencies to comply with the Public Records Law	Approved and Implemented

Architecture	Project Name	Project Description	Status as of 07/ 2000
		and the North Carolina Government Information Locator Service.	
Componentware	Component Library and Reuse Program	Establish a central repository for maintaining information about software components and services that can be shared by multiple application systems. Develop the methodology and framework for using the components.	Pending
Middleware	Service Broker and Message Oriented Middleware	Provide a communications infrastructure for application-to-application communications. The service broker will facilitate n-tier application design.	Implemented in 1/3 of state agencies and provided as an operational service
Groupware	Groupware Procurement	Expand state contracts to streamline the procurement process for groupware products such as imaging and document management system components.	Imaging and Document Management contracts pending. Email and Calender products contracts in place
	Enterprise Image Management	Develop common imaging services that are packaged as reusable components and made available statewide to application developers to: save standard file format images in image repositories record the location and ownership of the images in a central image index retrieve images for viewing.	Pending

Architecture	Project Name	Project Description	Status as of 7/2000

Groupware (continued)	Document Management Strategy	Conduct a study project to determine how, in the absence of industry standards, to cost-effectively store and share work-in-progress across departmental boundaries. Workflow and collaboration are key areas to be addressed.	Pending
Information	Federated Metadata Repository	Phase 1: Establish federated metadata definitions. Phase 2: Select, purchase, and implement a repository tool. Phase 3: Conduct a pilot project to populate the repository with information about federated metadata	Phase I/II/II Approved and Implemented
Integration	Interface Engine	Implement a vendor-supplied application integration middleware solution that streamlines the development of application interfaces when the applications themselves cannot be modified.	Implemented, in use, and available
Systems Management	Help Desk	Implement a statewide model for client support and help desk services which: Unifies the various help desks throughout the state to appear as a single help desk Facilitates the sharing of problem resolution information Employs software to support the help desk operations and provide management information Achieves economies of scale in the installation and use of help desk software.	Pending
	Distributed Systems Management	Provide an enterprise management infrastructure that uses an integrated systems management tool set and enables the centralized management of the state's complex, distributed technical infrastructure in a cost-effective manner.	Pending

Security and Directory	Public Key Infrastructure	Provide an enterprise public key infrastructure as a foundation for identification, authentication, authorization, non-repudiation, and confidentiality	Pilot Implementation project underway
	Directory Interoperability	Implement a directory interoperability solution to join disparate directories for the purposes of reduced administration and population of an authoritative source for directory based information	Project planning underway.

Architecture	Project Name	Project Description	Status as of 11/ 97
	Information Protection Policy	Define the state's information protection goals which include core policies and baseline controls for implementing and maintaining appropriate measures to safeguard the state's information resources. *Go to the project plan.*	Approved
Enabling Foundation	Quality Assurance	Further refine the IRM's quality assurance practices and procedures, and implement supporting tools, in order to ensure the technical integrity of system designs and the successful completion of applications development projects. *Go to the IRM Quality Assurance web page at www.state.nc.us/IRM/resource/qa/qahp.htm*	Approved and implemented
	Training	Execute the IRM training program to equip state business and technical personnel with the skills and knowledge necessary for implementing and using information systems that are compatible with the Statewide Technical Architecture. *Go to the IRM Education and Training web page at www.state.nc.us/IRM/training/overview.htm*	Approved
	Technology Planning	Further refine the existing procedures and requirements for the departmental technology planning process to ensure the coordination of technology and business planning efforts.	Approved

Table 1-3: Statewide Technical Architecture implementation initiatives